Selective Data Expansion for Model Performance

Jahid Hasan
Purdue University
West Lafayette, IN, USA
hasan89@purdue.edu

Romila Pradhan
Purdue University
West Lafayette, IN, USA
rpradhan@purdue.edu

ABSTRACT

Machine learning systems are increasingly being used in critical decision-making, such as healthcare, finance, and criminal justice. Concerns around system fairness have resulted in several mitigation techniques that emphasize the need for high-quality data to ensure fairer decisions. However, the role of earlier stages in machine learning pipelines in addressing model unfairness remains underexplored. We focus on the task of *selective data expansion*— carefully selecting additional data points from a data pool to add to the training data— to rapidly improve the fairness of a model learned on the modified data while also preserving model accuracy. Since not all points in the pool are equally beneficial, we propose DataSift, a data expansion framework that combines data valuation with multi-armed bandits to identify the most valuable data points for including into training data. Unlike prior methods that mitigate unfairness through data transformation or post-/in-processing, DataSift addresses the problem directly by selecting the right data. Over successive iterations, DataSift selects a partition, samples a batch of points leveraging influence functions, evaluates their impact, and updates partition utilities accordingly. Empirical evaluation of DataSift on multiple real-world and synthetic datasets shows that model unfairness is mostly resolved by including as few as 4% of additional data with at most 2.6% reduction (and as much as 27.4% increase) in accuracy.

KEYWORDS

Data Expansion, Algorithmic Fairness, Multi-Armed Bandits, Data Valuation

ACM Reference Format:
Jahid Hasan and Romila Pradhan. 2018. Selective Data Expansion for Model Performance. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 16 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

With the increasingly widespread use of machine learning (ML) in consequential decision-making domains, such as criminal justice, healthcare, and housing, there is an ever-growing need to ensure that ML-based systems do not have adverse implications on society. Designed carefully, these systems can potentially eliminate the unwanted aspects of human decision-making (e.g., biased decisions). However, irresponsible uses of artificial intelligence (AI) can lead to and reinforce systemic biases, discrimination and other abuses often reflected in the underlying training data [2, 10, 30]. A number of fairness metrics have been introduced over the past decade to quantify the discrimination exhibited by the ML-based systems [41, 61]. Simultaneously, the need to debias these systems has given rise to several bias mitigation techniques (see [14, 41] for recent surveys on fairness and bias in machine learning).

The focus on data-centric AI has spotlighted the importance of *data quality* in improving machine learning performance [51, 65–67]. In a recent survey, data science practitioners have reported feeling the most control over their data during earlier stages in the data science pipeline such as data collection and curation [32]. Several recent works have particularly recognized the significance of augmenting the training data from accessible data sources, termed as *data expansion*, for improving machine learning model performance [21, 39]. Data expansion differs from the task of data acquisition which focuses on identifying additional data that should be acquired from existing/new sources [15, 37, 38, 59]. Most of the existing works on data acquisition to enhance machine learning models focus on traditional performance metrics (e.g., model accuracy, loss) [15, 37, 59] or model confidence [38]. Each of these solutions is tailored to the specific performance metric (e.g., distance-based clustering for accuracy, sampling by decision boundary distance for confidence, power-law region [42] observing for loss) and is therefore not directly applicable to the equally important metric of model fairness where recent research has highlighted the importance of different stages of data science pipelines in combating fairness violations [8, 9].

The following example illustrates the need for adding useful additional data to mitigate unfairness in model decisions.

Example 1.1. Consider an organization that deploys an automated algorithm to predict whether an individual qualifies as high income—a critical step for determining eligibility for social benefits reserved for higher-income groups. Although the model demonstrates high predictive accuracy on historical data, it systematically underestimates the likelihood of high income for female applicants, despite qualifications comparable to those of male applicants. This results in a documented demographic disparity of nearly 17% in outcomes [41], raising serious concerns about fairness and equity in resource allocation. To address these inequities, a data scientist working in the organization considers expanding the training set with additional samples from the AdultIncome dataset [24] (see Table 2 and Section 4.1.1 for more details) with the same schema. However, the pool itself is demographically skewed: women are underrepresented overall, and especially so in the high-income category. As shown in Figure 1, simply adding the entire data pool perpetuates the same disparity. This behavior persists when the

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

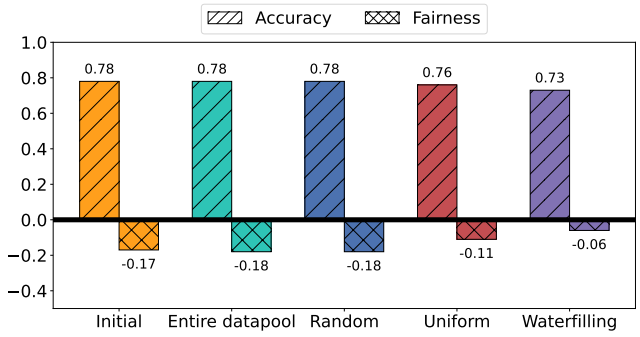


Figure 1: Impact of different data selection strategies on model fairness (statistical parity) and classification accuracy. Lower magnitude of fairness indicates a less biased model.

initial training dataset is expanded with 20% of data chosen randomly from the data pool. With an equal selection of data from the two demographic groups (Uniform), we observe a modest improvement in fairness but at the cost of accuracy. However, an even deliberate selection of data that enforces demographic balance in the training set (Waterfilling) results in a 65% improvement in fairness (but causes a sharp 6% decline in accuracy). These results highlight the inherent trade-off between fairness and accuracy, and the importance of selective data expansion that preserves both. (Indeed, Figure 3a shows that through strategic reordering of data selection, DataSift achieves substantial fairness improvements over the baselines while preserving accuracy, using only 8% of the data pool. Further details are provided in Section 4).

Disparate representation rates of demographic groups in training data are widely recognized as a primary source of fairness violations [3, 44, 53, 54]. As described in Example 1.1, while ensuring equal representation of demographic groups can partially improve fairness, it often comes at the expense of accuracy. This paper, therefore, addresses a central question: which additional data points must be added to the training data to improve fairness while simultaneously preserving the accuracy of the resulting model?

Given a data pool $\mathcal{D}$, our goal is to add a subset of points that maximally improves fairness while preserving accuracy. This can be cast as a *subset selection* problem, which is NP-hard in general [20]. We focus on determining the best subset that has up to K data points (typically, $K \ll |\mathcal{D}|$). However, not all such subsets improve fairness, and identifying the best one requires evaluating $\binom{|\mathcal{D}|}{k}$ possibilities ($k \in [1, K]$). This exhaustive search is prohibitively expensive due to the exponential number of subsets and the retraining cost per evaluation ($O(|\mathcal{D}|^K \times Tr)$, where Tr is the cost of retraining). An alternative solution constructs the subset by sequentially adding up to K *best data points* rather than finding the best subset. The naïve approach requires evaluating each data point in the data pool individually and selecting the top-K most valuable data points with an $O(|\mathcal{D}| \times Tr)$ complexity. To reduce the number of times the model is retrained, data are often evaluated in batches rather than one at a time. The challenge then is to construct batches such that we do not unnecessarily evaluate data points that do not improve fairness at all.

We propose DataSift, a framework that integrates data valuation with multi-armed bandits (MAB) [62], a special case of reinforcement learning [58], to *efficiently* evaluate the utility of data points based on their joint impact on model fairness and accuracy when incorporated into training. DataSift reduces the search space by systematically partitioning the data pool into smaller subsets and determining the order of incorporating into training data through a principled balance between **exploring** new partitions and **exploiting** influential data points within the selected partition. Partitioning can be performed automatically (e.g., clustering) or guided by domain knowledge (e.g., stratification by state, demographics, or timeframes). Related to our work, AutoData [15] proposed an MAB-based technique to selectively acquire data from heterogeneous data sources to enhance the accuracy of a learned model, but has limitations: (a) its reward formulation is explicitly accuracy-centric and does not account for the inherent trade-off between fairness and utility, making its direct application to fairness-aware data acquisition difficult; (b) within each partition, it acquires random batches, which limits systematic exploration of beneficial data points; and (c) it retrains a model to evaluate each such batch, rendering it inefficient for large datasets and complex models.

To address these limitations, DataSift employs an upper confidence bound (UCB) [4] strategy guided by a *reward score* that jointly accounts for both **accuracy** and **fairness** when selecting a partition for expansion. Within a selected partition, DataSift further leverages data valuation to prioritize candidate data points based on their estimated impact on model fairness, and selects the top-k points for acquisition. To ensure computational efficiency, we employ influence functions [17, 35] to approximate the impact of including individual data points on fairness metrics, thereby avoiding repeated full model retraining for each candidate batch.

Summary of contributions. Our main contributions can be summarized as follows:

- We formalize the problem of **selective data expansion for improving the fairness** of an ML model in classification tasks. (Section 2)
- We present DataSift, a system that casts the task of selective data expansion for model fairness as a **multi-armed bandit** (MAB) problem and solves it using the upper confidence bound (UCB) algorithm based on a novel reward score that addresses both model fairness and accuracy. (Section 3.1)
- To carefully construct a batch for addition, DataSift incorporates the concept of **data valuation** and leverages *influence functions* to estimate the importance of data points toward model fairness, which in turn speeds up DataSift. (Section 3.3)
- We conduct extensive experiments on six real-world datasets to demonstrate the effectiveness of DataSift in rapidly improving model fairness while preserving accuracy and present trade-offs between effectiveness and efficiency of the proposed methods. (Section 4)

2 PRELIMINARIES

This section formally defines the terminology and problems we addressed in this paper.

Classification. We consider a binary supervised learning task defined on data domain $\Gamma = \{\mathcal{X}, \mathcal{Y}\}$, where $\mathcal{X}$ denotes the feature

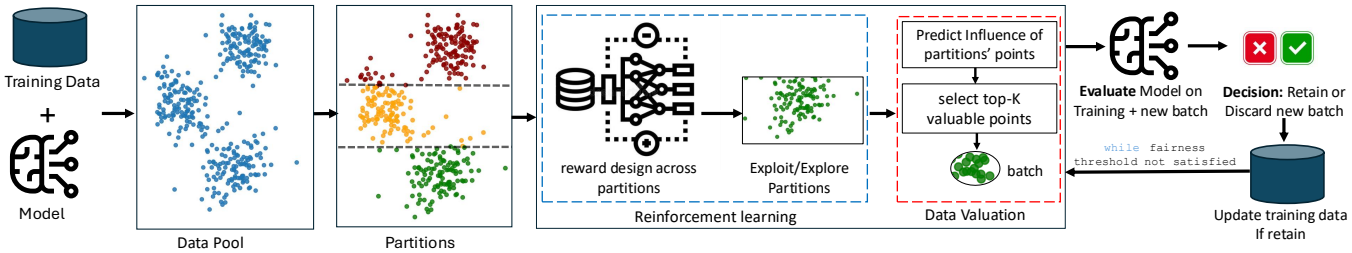


Figure 2: Overview of DataSift for selective data expansion to improve model fairness. Blue-dashed box highlights DataSift with Reinforcement Learning: multi-armed bandit (MAB) framework, while the red-dashed part incorporates data valuation.

space over p features and $\mathcal{Y} = \{0, 1\}$ denotes the binary label space. Suppose there is a conditional distribution $p(y \mid \mathbf{x})$ defined over Γ, where $\mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$. Given training dataset $\mathbf{D}_{train} = \{d_i\}_{i=1}^n = \{\mathbf{x}_i, y_i\}_{i=1}^n \in \Gamma$, the learning task is to train a classifier $\mathcal{M}$ that represents a distribution g that captures the target distribution p as closely as possible. $\mathcal{M}$ learns a function $f : \mathcal{X} \to \hat{Y}$ that associates each data point $\mathbf{x}$ with a prediction $\hat{y} = f(\mathbf{x}) \in \{0, 1\}$, and is evaluated on $\mathbf{D}_{test} \in \Gamma$. Learning algorithm f trains on $\mathbf{D}_{train}$ to learn the optimal parameters $\theta^* \in \mathbb{R}^p$ that minimize the empirical loss $\mathcal{L}(\mathbf{D}_{train}, \theta) = \frac{1}{n} \sum_{i=1}^n L(d_i, \theta)$.

Algorithmic group fairness. Given a binary classifier $\mathcal{M} : \mathcal{X} \to \hat{Y}$ and a protected attribute $S \in \mathcal{X}$ (such as gender, race, age, etc.), we interpret $\hat{Y} = 1$ as a favorable (positive) prediction and $\hat{Y} = 0$ as an unfavorable (negative) prediction. We assume the domain of S, $\text{Dom}(S) = \{0, 1\}$ where $S = 1$ indicates a privileged and $S = 0$ indicates a protected group (e.g., males and non-males, respectively). We select the widely popular setting of binary classification with binary protected attributes; however, solutions presented in this paper are easily extensible to settings for multi-class classification and intersectional fairness. Algorithmic group fairness mandates that individuals belonging to different groups must be treated similarly. The notion of similarity in treatment is captured by different associative notions of fairness such as statistical parity (a.k.a. demographic parity), predictive parity, and equalized odds [16, 41, 61]. For example, for the widely-used group fairness metric of statistical parity, model $\mathcal{M}$ satisfies statistical parity if both the protected and the privileged groups have the same probability of being predicted the positive outcome i.e., $P(\hat{Y} = 1|S = 0) = P(\hat{Y} = 1|S = 1)$. Although these commonly used metrics are tailored to binary classification, DataSift is not inherently tied to binary classification; instead, it relies on a data valuation signal and an acquisition policy that are agnostic to the label structure. Extending DataSift to multi-class or multi-label classification primarily requires redefining the fairness objective for a multi-label setting, for example, by aggregating class-wise fairness violations or applying one-vs-rest formulations. As such, the framework naturally generalizes to more complex prediction settings beyond binary classification, which is orthogonal to DataSift's mechanism. We the define chosen fairness metric by $\mathcal{F}$ and quantify fairness in the predictions over $\mathbf{D}_{test}$ by a model trained on $\mathbf{D}$ by $\mathcal{F}_\mathbf{D}$. For example, for statistical parity, $\mathcal{F}_\mathbf{D} = P(\hat{Y} = 1|S = 0) - P(\hat{Y} = 1|S = 1)$ quantifies the difference in the probabilities of protected and privileged groups having a positive outcome. If $\mathcal{F}_\mathbf{D} < 0$, the model is biased against the protected group while $\mathcal{F}_\mathbf{D} > 0$ indicates the model is biased against the privileged group. A higher value of $|\mathcal{F}_\mathbf{D}|$ indicates lower fairness (greater disparity) in the model's predictions.

Data Pool. In our framework, data pool D denotes an auxiliary collection of candidate examples that are available for potential acquisition but are not included in the initial training set. The pool may be constructed from multiple heterogeneous sources or from a single homogeneous source; in this work, we focus on the latter setting, where D is derived from a single source and thus homogeneous. In practice, such pools naturally arise from historical archives, held-out operational data, or newly arriving data in operational pipelines. Additionally, for tabular learning tasks, D can be formed through externally discovered relational datasets obtained from data lakes or Web APIs that expose tables with partial attribute overlap with the training data [26, 45, 46]. These datasets may require lightweight schema alignment, with missing attributes handled using standard techniques such as NULL imputation. For vision tasks, the pool can be formed from publicly available benchmarks or images retrieved through Web-based search APIs using task-relevant labels. The choice and composition of the data pool directly influence the expansion process, as they determine which candidate examples can be evaluated and selected. A more diverse pool increases the likelihood of identifying points that improve utility or mitigate fairness gaps, while a skewed or limited pool may restrict attainable gains. Importantly, our method does not assume that the pool follows the same distribution as the training data and remains effective under distributional shifts, as demonstrated in Section 4.4. The acquisition strategy relies on the observed contribution of candidate subsets rather than global distributional alignment. Finally, we assume a uniform cost for incorporating any point from the pool, consistent with scenarios where the pool has already been acquired.

Problem Definition. Given a model trained on $\mathbf{D}_{train}$, fairness metric $\mathcal{F}$, and data pool $\mathcal{D}$, we address the problem of determining additional data points $\mathbf{D}_{exp} \subset \mathcal{D}$ that must be added to $\mathbf{D}_{train}$ such that the model learned on $\mathbf{D}_{train} \cup \mathbf{D}_{exp}$ is fairer than the original model learned on $\mathbf{D}_{train}$ alone (i.e., $|\mathcal{F}_{\mathbf{D}_{train} \cup \mathbf{D}_{exp}}| < |\mathcal{F}_{\mathbf{D}_{train}}|$).

3 DATA EXPANSION FRAMEWORK

In this section, we introduce DataSift, a framework for acquiring data points that enhance the fairness of a learned model. Section 3.2 formulates the task as a multi-armed bandit (MAB) problem, which identifies the partition with the highest potential for fairness improvement but not the specific points to select. To address this, Section 3.3 introduces a data valuation-based batch selection strategy for targeted acquisition. By integrating these components, DataSift jointly improves fairness and overall model performance.

3.1 Multi-Armed Bandit (MAB) framework

The Multi-Armed Bandit (MAB) [62] maps a framework for sequential decision-making under uncertainty. This problem can be framed by the metaphor of a gambler (or 'Agent') choosing which of several slot machines (or 'Arms') to play in each attempt to maximize the total prize over a series of trials. Considering that gamblers have some knowledge about each slot machine from initial trials, they are faced with the question of which machine to select next. The multi-armed bandit framework is ideal for solving this dilemma. MAB can formally be defined as: at each state t, the agent selects an arm i from a set of K available arms and receives a reward r_t from a distribution associated with that arm, which is generally unknown to the agent. MAB aims to determine which arm to pull next to maximize the cumulative reward R after T rounds, guided by the principle of balancing *exploration* and *exploitation*.

Exploration involves choosing unexplored options to gather new information about their potential rewards. By exploring, the agent reduces uncertainty about less-known arms, potentially uncovering actions that provide higher rewards than initially expected. On the other hand, *exploitation* involves selecting the action that currently offers the highest reward based on the agent's existing knowledge. While exploitation maximizes short-term gains by considering known information, it may lead to suboptimal long-term outcomes if the agent overlooks better options that have not been sufficiently explored. At the same time, excessive exploration may waste resources on testing suboptimal actions, thus missing opportunities for immediate reward maximization.

Thus, the agent must balance between exploring and exploiting to maximize cumulative rewards eventually. To find a trade-off between exploration and exploitation, making optimal short-term decisions based on available information is crucial to solving this dilemma. This trade-off is central to various algorithms, such as Thompson sampling [1], ϵ-greedy[36], and Upper Confidence Bound (UCB) [5] designed to maximize the agent's long-term performance. Due to its computational flexibility, we adopt the UCB algorithm for our problem, while noting that alternative algorithms could also be applied to DataSift. Similar to other base algorithms for MAB, the UCB algorithm does not always yield exact optimal results, but its performance is near-optimal [11].

Next, we will map the problem of data expansion for improving model fairness to the MAB framework and discuss our approach.

3.2 Mapping data expansion for fairness to MAB

To cast the data expansion problem to the MAB framework, we first split the data pool into several disjoint partitions, i.e., $\mathcal{D} = \bigcup_{i=1}^{g} C_i$ where g is the number of partitions. These partitions could be obtained by clustering $\mathcal{D}$ using existing clustering algorithms such as multivariate Gaussian Mixture Model (GMM) [27], k-means [33], hierarchical [43], and DBSCAN [25] or partitioning the data pool based on some criterion or by considering the data pool as a collection of data sources. Partitioning the data pool offers two advantages: i) ensures systematic exploration of overlooked data composition critical for fairness, ii) and it reduces computational complexity by restricting selection to smaller, structured subsets rather than the entire data pool. However, each partition C_i is then treated as an *arm* in the multi-armed bandit (MAB) framework. In the k-th iteration, the algorithm selects a partition C_k and samples a batch from C_k for evaluation. We will discuss the batch selection process in more detail in Section 3.3. Subsequently, the selected batch is merged with the current training dataset, and the fairness of the resultant model trained on the updated training dataset is reported. The change in the model fairness before and after adding the batch determines whether the batch should be retained or discarded and whether the selected partition is rewarded or penalized. The reward/penalty score is utilized to select the subsequent partition to evaluate more data points. The blue-box part in Figure 2 illustrates the data expansion task, framed as a multi-armed bandit problem.

Reward/Penalty score. The algorithm scores the chosen partition according to the change in model fairness after merging the batch with existing training data. The reward for partition C_i at iteration k is defined as r_i^k. If fairness improves, the partition is rewarded, otherwise penalized. However, in addition to allocating reward/penalty scores to the chosen partition, the *full feedback* [57] MAB algorithm also assigns a portion of scores to other partitions that could have been selected. This approach efficiently accelerates the process by reducing the number of evaluations. Several studies on MAB-based data expansion for improving model performance [15, 63] consider the distance between partitions when assigning rewards or penalties to other partitions. The intuition behind this scoring is based on the assumption that partitions that are closer to the selected partition have a higher likelihood of getting selected and, hence, should be rewarded similarly. This assumption holds for performance metrics such as accuracy and confidence because closer partition centroids indicate that the partition shares similar characteristics. However, this setting might not hold for model fairness because partitions that are close might have extremely different compositions over sensitive attributes and, therefore, might impact fairness differently. In other words, the *base rate difference* of the partition plays a significant role in fairness. Recall that the base rate difference for dataset D is defined as: $\Delta BR_D = P(Y = 1 \mid S = 0) - P(Y = 1 \mid S = 1)$. In the context of fairness, the selection of a partition indicates an improvement in model fairness as a result of inherent lower base rate difference among the different demographic groups in the partition. Consequently, other partitions with similar base rates should be rewarded higher than those with worse base rates. In fact, we show that incorporating intra-partition base rate differences among the different demographic groups is much more effective in the proper distribution of the reward among partitions, resulting in significantly improved fairness (see Section 4.4 for more details).

Focusing solely on the base rate difference, however, can degrade the overall accuracy of the learned model. We, therefore, propose a *novel reward score* that caters to both fairness and accuracy with a balance parameter to split the reward scores among the different partitions. When partition C_i is evaluated, the reward score for each partition C_j is computed as follows:

$$r_j = \frac{\Delta\mathcal{F}}{1 + |\Delta BR_{C_j}|} + \beta \frac{\Delta acc}{1 + dist(C_i, C_j)} \tag{1}$$

where $\Delta\mathcal{F} = \mathcal{F}_{\mathbf{D}_{train}^{k-1} \cup b_i} - \mathcal{F}_{\mathbf{D}_{train}^{k-1}}$ denotes the change in model fairness after adding batch b_i to training data up to $k-1$ iterations denoted by $\mathbf{D}_{train}^{k-1}$, while Δacc stands for accuracy change. The

parameter β is used to balance the contributions of fairness and accuracy in the reward scores. Additionally, ΔBR_{C_j} indicates the intra-partition base rate difference in C_j and $dist(C_i, C_j)$ is the normalized Euclidean distance between the two partitions computed over the partition centroids. A positive value in r_j incurs a reward, while a negative value incurs a penalty.

Aggregated reward/penalty score. The multi-armed bandit approach aggregates the prior reward and penalty information up to k iterations to make an informed decision in the next iteration. Let R_i^k represent the aggregate score of partition C_i from iteration 1 to k, defined as: $R_i^k = \frac{1}{n_i^k} \sum_{j=1}^{k} r_i^j$, where r_i^j represents the reward score of C_i at the j-th iteration, and n_i^k denotes the number of times C_i is selected and rewarded a positive score up to the k-th iteration.

Upper Confidence Bound (UCB). The UCB algorithm is designed to adaptively adjust the trade-off between exploration and exploitation over time [5], which is achieved by incorporating a measure of uncertainty or confidence in the estimated rewards of each arm. This measure is used to guide the decision-making process, allowing the algorithm to explore arms with potentially high but uncertain rewards while also exploiting arms with known high rewards. Due to its deterministic nature, it has been widely used in the field of MAB, Reinforcement Learning, and Recommendation Systems [4, 13, 28, 48]. We determine the UCB score for partition i in the k-th iteration as:

$$U_i^k = R_i^k + \alpha \sqrt{2 \ln \left(\frac{n^k}{n_i^k + 1} \right)} \tag{2}$$

where α is a pre-defined parameter that maintains the balance between exploration and exploitation and $n^k = \sum_{i=1}^{g} n_i^k$ for the total number of partitions g. The first term in Equation 2 represents exploitation while the later pertains to exploration. A partition with a higher reward will have a higher exploitation score, whereas one selected less frequently will have a higher exploration score.

3.2.1 Role of Batch Selection. MAB plays a central role in identifying the partition with the highest expected reward. Once a partition is selected, however, sampling a batch from it is non-trivial: while the partition may be promising overall, not every randomly chosen subset will improve model performance. AutoData [15] adopts a random batch selection strategy, but over time, this approach performs no better than naive random acquisition from the entire pool—even when combined with MAB (see results in Figure 3).

To facilitate more effective batch selection, the following section introduces data valuation approach for enhanced batch selection.

3.3 Valuation-based Batch Selection

To carefully construct a batch to merge from a chosen partition, we leverage the idea of data valuation, which has been successfully used in explainable AI [29, 35] to quantify the contribution of training data points toward the performance of the learned model. Due to their effectiveness in accurately estimating the contribution of data points and faster online computation time, we use first-order *influence functions* [17, 35] to approximate the importance of training data points toward model fairness. Based on the computations, we construct the batch to include the chosen partition.

3.3.1 Influence functions. Recall from Section 2 that θ^* is the set of optimal parameters that minimize the empirical risk, i.e.,

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\ \mathcal{L}(\theta) = \underset{\theta \in \Theta}{\operatorname{argmin}}\ \frac{1}{n} \sum_{i=1}^{n} L\left(d_i, \theta\right) \tag{3}$$

To incorporate influence functions, we assume that the empirical risk function $\mathcal{L}(\theta)$ is twice-differentiable and strictly convex. Under these conditions, we can guarantee the Hessian matrix $\mathcal{H}_\theta$ exists and is positive definite, and therefore, its inverse $\mathcal{H}_\theta^{-1}$ also exists. These assumptions are applicable to a wide range of classification algorithms such as logistic regression, support vector machines, and feed-forward neural networks.

Let $\nabla_\theta \mathcal{L}(\theta)$ and $\mathcal{H}_\theta = \nabla_\theta^2 \mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta^2 L\left(d_i, \theta\right)$ be the gradient and the Hessian of the loss function, respectively. The influence of up-weighting a data point $\mathbf{d} \in \mathbf{D}_{train}$ by ϵ on the model parameters is computed as:

$$I_\theta(\mathbf{d}) = \left. \frac{d\theta_\epsilon^*}{d\epsilon} \right|_{\epsilon=0} = -\nabla_\theta^2 \mathcal{L}\left(\theta^*\right)^{-1} \nabla_\theta L\left(\mathbf{d}, \theta^*\right) = -\mathcal{H}_\theta^{-1} \nabla_\theta L\left(\mathbf{d}, \theta^*\right) \tag{4}$$

To add a data point $\mathbf{d}$ to training data, we up-weight it by $\epsilon = \frac{1}{n}$. Therefore, the influence of $\mathbf{d}$ on model parameters can be linearly approximated by computing $d\theta_\epsilon^* \approx \frac{1}{n} I_\theta(\mathbf{d})$.

Using the chain rule of differentiation, we can estimate the effect of up-weighting data point $\mathbf{d}$ on any function $f(\theta)$ as:

$$\mathcal{I}_f(\mathrm{d}) = \left. \frac{df\left(\theta_\epsilon^*\right)}{d\epsilon} \right|_{\epsilon=0} = \left. \frac{df\left(\theta_\epsilon^*\right)}{d\theta} \frac{d\left(\theta_\epsilon^*\right)}{d\epsilon} \right|_{\epsilon=0} = \nabla_\theta f\left(\theta^*\right)^\top \mathcal{I}_\theta(\mathrm{d}) \tag{5}$$

When $f = \mathcal{F}$, we obtain the first-order influence of a single training data point $\mathbf{d}$ on model fairness by approximating $d\mathcal{F}_\epsilon^* \approx \frac{1}{n} \mathcal{I}_\mathcal{F}(\mathrm{d})$.

3.3.2 Batch construction. Given a data point $\mathbf{d} \in \mathbf{D}_{train}$, the influence function approximation in Equation 5 estimates the effect of up-weighting $\mathbf{d}$ on model fairness. To generalize this notion to the unlabeled data pool $\mathcal{D}$, we first compute first-order influence estimates on $\mathbf{D}_{train}$ and train a polynomial Ridge regression model to approximate the fairness influence as a function of data features. The regressor is trained using features from D_{train}, with targets given by the estimated fairness influence obtained via influence functions. The learned regression model $\mathcal{R}$ is then applied to unseen data points in each partition $C_i \subset \mathcal{D}$ to predict their expected fairness influence on the current model $\mathcal{M}$. Data points within each partition are subsequently ranked in descending order of predicted influence, as formalized in `SortPartitions()`. At each iteration, a batch b is constructed by selecting the top-K data points with the highest predicted influence on fairness.

Function `SortPartitions`$(C = \{C_1, \ldots, C_g\}, \mathbf{D}_{train}, \mathcal{M})$**:**
  $I \leftarrow$ GET_INFLUENCE$(\mathbf{D}_{train}, \mathcal{M})$ as in Eq. 5
  $\mathbf{D}_{train} \leftarrow \{(x, y, I(x)) : (x, y) \in \mathbf{D}_{train}\}$
  $\mathcal{R} \leftarrow$ TrainRegressor$(\mathbf{D}_{train})$
  for $i \leftarrow 1$ **to** g **do**
    for $x \in C_i$ **do**
      $\hat{I}(x) \leftarrow \mathcal{R}(x)$
    $Sorted\_C[i] \leftarrow \mathrm{Sort}\left(C_i; \hat{I}(x) \text{ desc.}\right)$
  return $Sorted\_C$

Note that a batch $b \subseteq \mathcal{D}$ constructed in the above manner does not explicitly account for the inherent correlations between data points in b. As such, the estimated influence of the batch computed simply by adding the first-order influences of individual data points might not be exact, i.e., $\mathcal{F}_{\mathcal{D}\cup b}$ is not necessarily equal to $\mathcal{F}_{\mathcal{D}} + \frac{|b|}{n}\sum_{\mathbf{d}\in b}\mathcal{I}_{\mathcal{F}}(\mathrm{d})$. Therefore, the selected batch is not guaranteed to be optimal. Identifying the optimal batch within a selected partition can be formulated as a *subset selection* problem, which is NP-hard [20]. In practice, however, employing influence function approximations as a heuristic for batch construction has been shown to yield near-optimal performance (see Section 4.3).

3.4 DataSift Framework

In this section, we present DataSift, a data expansion framework designed to enhance model performance. Figure 2 provides an overview. At a high level, given a biased model trained on an initial dataset and an available data pool, DataSift first systematically partitions the pool into smaller subsets to reduce the search space. It then employs a multi-armed bandit (MAB) strategy to identify the partition with the highest potential reward (blue-dashed box in Figure 2). In each round, DataSift applies data valuation to extract the top-K influential data points to form a batch from the selected partition (red-dashed part in Figure 2). The model is retrained with the newly acquired batch and re-evaluated. Based on the observed performance, DataSift decides whether to retain the batch—updating the training data—or discard it. This process iterates until the stopping criteria are met, ultimately yielding a fair model trained on the updated data.

Early stopping criteria. The Algorithm 1 is designed to acquire data points from the available pool in order to maximize improvements in model fairness. In practice, however, we observe that the model often achieves near-fairness (i.e., a parity difference close to zero) after incorporating only a fraction of the pool. Continuing to evaluate and add further data in such cases leads to unnecessary computational overhead. Moreover, if the entire pool were added, all methods would eventually converge to the same performance, obscuring the benefit of selective expansion. To address this, we define an expansion budget, B, representing a fraction of the data pool, and introduce a fairness threshold, τ, which serves as an early stopping criterion in Algorithm 1. The expansion process is halted and the data points added so far are returned if: (a) model fairness is within threshold τ, or (b) expansion budget B is exhausted.

Algorithm 1 provides the pseudocode of DataSift, which is designed to enhance fairness without degrading accuracy. The algorithm takes as input the set of partitions, training and test data, the base model, the fairness threshold τ, the batch size K, and the expansion budget B, and outputs the acquired subset of data points. **Lines 1–3:** The procedure begins with the initialization of key variables and parameters. **Lines 4–17:** The main loop iterates until either the specified fairness threshold is reached or the expansion budget is exhausted. **Lines 6–8:** In each iteration, the partition with the maximum UCB score is selected, from which a batch of size K with the highest predicted influence values is sampled and subsequently evaluated. **Lines 9–14:** The sampled batch is incorporated into the training data only if it leads to an improvement in fairness relative to all previous iterations, after which the training set and the corresponding partition are updated. **Lines 15–17:** The algorithm then updates the reward and penalty values for each partition, along with the aggregated and UCB scores. Finally, the updated training set is returned.

Algorithm 1: DataSift Framework

Input: $C = \{C_1, \ldots, C_g\}, \mathbf{D}_{train}, \mathbf{D}_{test}, \mathcal{M}, \tau, K, B$
Output: $\mathbf{D}_{exp}$

1 $Sorted\_C \leftarrow$ `SortPartitions`$(C, \mathbf{D}_{train}, \mathcal{M})$;
2 **for** $i \leftarrow 1$ **to** g **do**
  $R_i, n_i, U_i \leftarrow 0$
3 $\mathbf{D}_{exp} \leftarrow [\,]$;
  $k \leftarrow 0$;
  $\mathbf{D}^0_{train} \leftarrow \mathbf{D}_{train}$;
  $\mathcal{F} \leftarrow \mathcal{F}(\mathbf{D}_{train})$;
4 **while** $|\mathcal{F}| \geq \tau$ ***and*** $B > 0$ **do**
5 $k \leftarrow k + 1$;
6 $i \leftarrow \arg\max_j U^k_j$;
7 sample $b_i \subset Sorted\_C[i]$, $|b_i| = K$;
8 $\Delta f \leftarrow \mathcal{F}(\mathbf{D}^{k-1}_{train} \cup b_i) - \mathcal{F}(\mathbf{D}^{k-1}_{train})$;
9 **if** $|\Delta f| > 0$ ***and*** $|\mathcal{F}(\mathbf{D}^{k-1}_{train} \cup b_i)| \leq |\mathcal{F}|$ **then**
10 $\mathbf{D}_{exp} \leftarrow \mathbf{D}_{exp} \cup b_i$;
11 $\mathbf{D}^k_{train} \leftarrow \mathbf{D}^{k-1}_{train} \cup b_i$;
12 $Sorted\_C[i] \leftarrow Sorted\_C[i] \setminus b_i$;
13 $\mathcal{F} \leftarrow \mathcal{F}(\mathbf{D}^k_{train})$;
14 $B \leftarrow B - K$;
15 **for** $j \leftarrow 1$ **to** g **do**
16 $r_j \leftarrow \dfrac{\Delta f}{1 + |\Delta BR_{C_j}|} + \beta \dfrac{\Delta acc}{1 + dist(C_i, C_j)}$;
17 **update** R^k_j, n^k_j, U^k_j;
18 **return** $\mathbf{D}_{exp}$

Algorithm 1 has a computational complexity of $O(I\times(|C|+E+t))$ where I denotes the number of required evaluations (iterations), $|C|$ is the maximum partition size, E is the evaluation complexity for each batch, and t refers to the complexity of score calculation. The maximum number of evaluations, I, is bounded by $I = A + R \leq \frac{N}{K}$, where A and R represent the number of accepted and rejected batches, respectively, N is the size of the data pool, and K is the batch size. Moreover, $A \leq \frac{B}{K}$ where B is the expansion threshold .

Note that the classical formulation of influence functions to compute the valuation of data points limits the applicability of the approach to parametric models with convex and twice-differentiable loss functions. While DataSift is instantiated using parametric models with convex, differentiable losses to enable efficient influence estimation, this assumption is not fundamental to the framework. The key requirement is the relative ranking of data utility, which is all DataSift requires for acquisition decision. In practice, influence-based estimates can be extended beyond this setting using standard approximations such as local second-order smoothing, damped Hessian inverses, or surrogate losses, which preserve relative data utility rankings [35]. Besides, prior work on data valuation has shown that Data Shapley approximations [29] can provide

	C_1		C_2		C_3		C_4		
Evaluation	$\Delta\mathcal{F}$	U_1	$\Delta\mathcal{F}$	U_2	$\Delta\mathcal{F}$	U_3	$\Delta\mathcal{F}$	U_4	**Decision**
1	-0.0118 ↓	-0.0118		-0.0073		-0.0068		**-0.0050**	$\Delta\mathcal{F}$ ↓–Discard batch, Next Cluster → C4
2		0.0393		0.0382		0.0373	0.0366 ↑	**0.0991**	$\Delta\mathcal{F}$ ↑–Retain batch, Next Cluster → C4
3		0.0595		0.0537		0.0510	0.0043 ↑	**0.0976**	$\Delta\mathcal{F}$ ↑–Retain batch, Next Cluster → C4
4		**0.0800**		0.0712		0.0673	0.0110 ↑	0.0800	$\Delta\mathcal{F}$ ↑–Retain batch, Next Cluster → C1
5	0.0050 ↑	0.1257		**0.1871**		0.1819		0.0999	$\Delta\mathcal{F}$ ↑–Retain batch, Next Cluster → C2

Table 1: Example breaks down the cluster and batch selection decisions for Algorithm 1 over the ACSIncome dataset, data pool size $|\mathcal{D}| = 50$. $\Delta\mathcal{F}$ represents the fairness metric change each round; U denotes the UCB score. Dark green indicates improvement in fairness after adding the selected batch; red indicates a decline. Bold value of U marks which cluster is selected next.

model-agnostic utility estimation, albeit at a higher computational cost. Thus, the parametric setting adopted in this work represents a practical and efficient instantiation of DataSift rather than a strict limitation. We also evaluate a model-agnostic variant of DataSift in Section 4, denoted as DataSift_M, which omits data valuation during expansion and instead relies on random batch sampling.

Table 1 presents a toy example illustrating the decision-making process of DataSift on a small data pool ($|\mathcal{D}| = 50$) from the ACSIncome dataset [23]. The pool was partitioned into an optimal number of clusters, and with a mini-batch size of two instances, the table reports the first five of 25 evaluations. It highlights how DataSift selects clusters and batches while balancing exploration and exploitation. Overall, in this small example, DataSift reduced model unfairness by 60%.

4 EXPERIMENTAL EVALUATION

This section presents experiments that evaluate the effectiveness of DataSift. We seek to answer the following research questions: **RQ1:** How effective is DataSift compared to existing methods in selecting additional data points to improve model fairness across different machine learning models? **RQ2:** What is the benefit of incorporating MAB and data valuation in DataSift? **RQ3:** How effective is DataSift with respect to the different hyperparameters and design choices? **RQ4:** How efficient are our different solutions with respect to varying dataset sizes?

4.1 Experimental Setup

4.1.1 Datasets. We consider several real-world datasets popular in the fair machine learning literature. **AdultIncome [24]** dataset used to predict whether an individual's annual income exceeds $50,000 by analyzing a range of demographic and socio-economic factors, including several sensitive attributes, such as age, sex, and race. **Credit [18]** dataset used to forecast the likelihood of delaying credit payments using financial and demographic history. American Community Survey (ACS)-based datasets [23] that provide a suite of datasets (including **ACSIncome, ACSPublicHealth, ACSMobility**, and **ACSEmployment**) for predicting different outcomes such as income level, public health status, mobility information, and employment status. The ACS datasets are highly skewed toward California (CA) state. Unless otherwise specified, our analysis primarily focuses on evaluating CA data for the year 2018. Table 2 summarizes the datasets, all of which exhibit demographic imbalance. Such imbalance is common in real-world applications and creates inherent challenges for fair model training, making them well-suited for evaluating our proposed approach.

4.1.2 Competing methods. We compared our proposed approach with several algorithms suited for data expansion:

Random. This naïve baseline method randomly selects a batch from the data pool in each iteration.

Entropy. This method selects the B data points with the highest predictive uncertainty [55]. Entropy of data point d_i is calculated as: $H(d_i) = -\left(p_i \log_2(p_i) + (1-p_i)\log_2(1-p_i)\right)$ where p_i is the probability that d_i is predicted by the model to have a positive outcome. A higher entropy value indicates that the model is less confident about its prediction for the data point. In each round, it selects the top-K data points with the highest entropy values.

AutoData [15]. This method uses the MAB framework for data acquisition to improve model accuracy. For comparison, we transform the constraint of this algorithm from accuracy to fairness.

SliceTuner [59]. This method fits per-slice loss–data curves and solves a budget-constrained convex program that balances accuracy and fairness by penalizing slices with above-average losses. For comparison, we consider observing the fairness metric and providing slices into four groups: the cross product of the binary sensitive attribute and the binary target outcome.

DataSift. This method represents our proposed solution described in Algorithm 1 that integrates the multi-armed bandit approach with data valuation.

DataSift_M. This method is a variant of DataSift that relies solely on the multi-armed bandit(MAB) approach, where batches are selected at random instead of data valuation.

Inf. This method focuses solely on data valuation–computes the influence of data points in the training data using Equation 5 and learns a regression model to estimate the influence of points in the data pool on model fairness. Each iteration selects the top-K data points with the calculated highest influence.

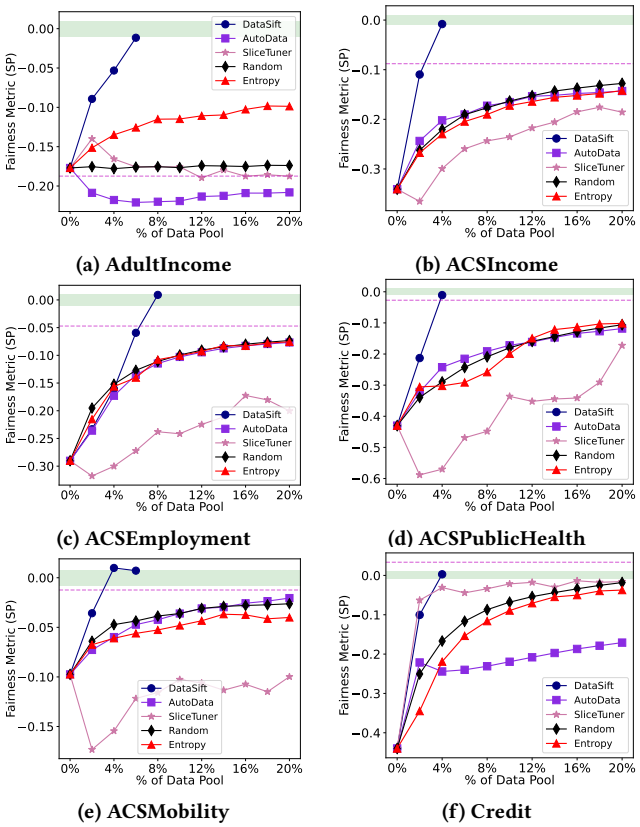
4.1.3 Fairness metrics. There are several metrics used to assess the fairness of a trained model, including *statistical parity*, *true positive rate*, and *predictive parity* [16, 41, 61]. Our algorithm effectively addresses any of these fairness metrics. Unless stated otherwise, all experiments are conducted using statistical parity, with some results reported under true positive rate parity and predictive parity.

4.1.4 Settings. We divided the entire dataset into three parts: $\mathbf{D}_{train}$ (Training), $\mathbf{D}_{test}$ (Test), $\mathbf{D}_{val}$ (Validation) and $\mathcal{D}$ (Data pool) in the ratio of $1:2:2:10$. All bandit decisions, fairness checks, and stopping criteria are now computed on D_{val}, while D_{test} is used only once at the very end to report accuracy and fairness. While test and data pool were split randomly, the training dataset was sampled to obtain an initial biased model. We partition the data

Table 2: Summary of datasets.

Dataset	Adult Income	Credit	ACSPublicHealth	ACSMobility	ACSEmployment	ACSIncome
Size	45, 222	150, 000	138, 554	80, 329	378, 817	250, 847
No. of features	8	10	19	22	16	11
Protected Group (S=0)	Female	Age < 35	Female	Afr.-Am.	Afr.-Am.	Female
Population of (S=0) (%)	33	12.8	55.9	5.2	4.8	47.2
Positive Labels in (S=0) (%)	11	11	35	73	39	34
Predictive Task	income> $50K?	serious delay in 2 years?	has public insurance coverage?	moved address last year?	employed?	income> $50K?

pool using Gaussian Mixture Model (GMM) [27] and find the optimal number of partitions ranges between two and seven using the Bayesian Information Criterion (BIC) [47]. Note that DataSift is similarly effective to the other choice of clustering algorithm. We demonstrate its performance with multiple methods, including k-means [33] and DBSCAN [25], Hierarchical-BIRCH [68] as detailed in Appendix A.2.2. The model was trained over $\mathbf{D}_{train}$ and evaluated on $\mathbf{D}_{test}$. The expansion budget B was set to 20% of the data pool, $|B| = .2 * |\mathcal{D}|$ and the batch size, $|K| = 10\%$ of the B. The target fairness threshold τ was set at 0.01. The trade-off parameter between exploration and exploitation (α in Equation 2) was set to 0.1 as [15]. The balance parameter β (in equation 1) was heuristically fixed at 1. We considered three ML algorithms: logistic regression, a support vector machine, a feed-forward neural network with one layer and ten nodes. We used the PyTorch [49] or sklearn [50] implementation of these algorithms.



(a) AdultIncome (b) ACSIncome

(c) ACSEmployment (d) ACSPublicHealth

(e) ACSMobility (f) Credit

Figure 3: Comparing DataSift with baselines to highlight the effectiveness in achieving fairness. The green shaded area represents the defined zone of unfairness.

4.2 Effectiveness of DataSift

In this set of experiments, we address RQ1 by comparing the performance of DataSift with four competing methods: Random, Entropy, SliceTuner, and AutoData. Unless otherwise specified, all methods are evaluated using Logistic Regression across all datasets. Figure 3 summarizes the results, where the x-axis denotes the fraction of the data pool acquired and the y-axis reports the fairness metric. Values near zero correspond to a fair model: negative values indicate bias against the protected group, while positive values indicate bias favoring the protected group. We define the fairness region as $|\mathcal{F}| < \tau = 0.01$, highlighted in green, with each marker in the figure corresponding to the acquisition of a batch.

The baselines Random and Entropy do not incorporate fairness considerations, instead selecting data points without leveraging prior information. Their trajectories are therefore similar across datasets: while adding more points yields marginal fairness improvements relative to the initial model—primarily because of the addition of new data points— they consistently fail to meet the fairness threshold.

AutoData leverages prior information for partition selection but samples batches randomly within partitions. Even when updated with a fairness constraint, it fails to improve fairness due to its distance-centric reward formulation. Consequently, its performance mirrors random selection in most cases. This limitation is most evident in AdultIncome and Credit (Figures 3a and 3f), where the proportion of positive labels is low (Table 2). In these settings, random, fairness-agnostic selection from partitions worsens fairness relative to others. effectively injecting additional bias.

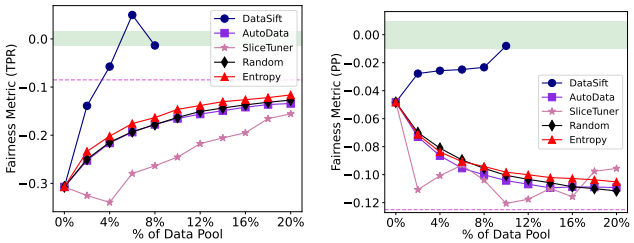
Adapting SliceTuner for fairness is non-trivial, as its design relies on power-law distributions. For completeness, we configured it with four slices defined by the cross-product of a binary sensitive attribute and a binary target attribute. While this reduces training loss, it fails to improve fairness across datasets. As shown in Figure 3, SliceTuner consistently suffers throughout the expansion process, highlighting that demographic-representation–based expansion alone is insufficient. An exception arises in **Credit** (Figure 3f), where the data composition allows SliceTuner to acquire more protected-positive samples, leading to incidental fairness gains as a byproduct of its loss minimization.

In contrast, DataSift integrates MAB-based partition selection with data valuation to identify fairness-influential data points within each partition. As illustrated in Figure 3, DataSift consistently achieves superior performance across all datasets, eliminating unfairness entirely while outperforming all competing methods at nearly every iteration. The sole exception is **ACSEmployment** (Figure 3c), where DataSift shows only marginal gains during the first two iterations; once partition selection changes, however, it rapidly identifies near-optimal batches and surpasses all baselines.

Overall, DataSift attains zero unfairness while using only a small fraction of the data pool. Specifically, it requires just 4–6% of the data to produce a fully fair model, whereas competing methods consume nearly 20% without reaching the fairness threshold, ultimately converging to the naïve adding full data pool baseline (dashed magenta line). Adding the entire data pool is computationally costly and still fails to meet the fairness threshold in all cases. By contrast, DataSift achieves both fairness and accuracy with far less data, demonstrating its efficiency and effectiveness. Note that DataSift adopts a *plug-and-play* design for the choice of fairness metric $\mathcal{F}$. Beyond statistical parity, we evaluate DataSift under true positive rate (Figure 4a) and predictive parity (Figure 4b) on the ACS Income dataset, with additional results reported in Appendix A.1. In all the cases, DataSift outperforms the baselines and reaches the target fairness threshold more efficiently.

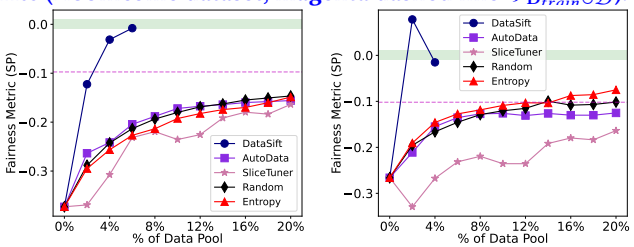
Model Generalization. The reliance on influence functions limits the applicability of DataSift to parametric models. To evaluate robustness within this scope, Figure 5 reports results on the **ACSIncome** dataset for two additional models: Support Vector Machines (SVM) and Neural Networks. For SVM (Figure 5a), DataSift requires one additional batch compared to Logistic Regression (Figure 3b), yet the overall trajectory and outcome remain closely aligned. The neural network case (Figure 5b) is more dynamic: the model initially shows bias toward the protected group, but after the first batch acquisition, the fairness metric shifts toward the privileged group. In this scenario, DataSift adapts effectively, making informed acquisition decisions that progressively mitigate bias and ultimately yield a fair model. These findings highlight the robustness of DataSift in correcting unfairness regardless of its direction or the demographic group affected, underscoring its versatility across parametric models. To extend applicability further, model-agnostic valuation methods such as Data Shapley [29] could be incorporated, enabling generalization to non-parametric models.

Effect on model accuracy. To improve fairness, algorithms prioritize diversifying the training set by adding data from partitions



(a) True Positive Rate (b) Predictive Parity

Figure 4: Evaluation of DataSift over different fairness metrics (ACSIncome dataset, magenta dashed line: $\mathcal{F}_{D_{train} \cup \mathcal{D}}$).



(a) Support Vector Machine (b) Neural Network

Figure 5: Evaluation of DataSift across parametric models.

that capture under-represented subgroups. This prevents the model from overfitting to privileged groups while reducing bias against marginalized ones. However, as Table 2 indicates, most datasets are dominated by the privileged group; thus, solely fairness-focused acquisition can risk accuracy degradation. Our reward design explicitly balances this trade-off, enabling DataSift to maintain accuracy while achieving fairness. Table 3 reports the final accuracy after data acquisition. DataSift often improves accuracy relative to baselines and, in datasets such as **AdultIncome** and **Mobility**, preserves accuracy within an acceptable tolerance. Methods such as Entropy and SliceTuner, which prioritize reducing model uncertainty, achieve slightly higher accuracy but at the expense of fairness, while AutoData suffers accuracy degradation as its constraint is modified to account for fairness.

Dataset	Methods					
	Initial	Random	Entropy	AutoData	SliceTuner	DataSift
AdultIncome	0.80	0.78	0.77	0.71	**0.80**	0.78
ACSIncome	0.70	**0.78**	0.77	0.78	0.77	0.77
Employment	0.69	0.75	**0.75**	0.74	0.74	0.73
PublicHealth	0.65	**0.69**	0.68	0.68	0.68	0.68
Mobility	0.77	0.77	**0.77**	0.77	0.75	0.75
Credit	0.73	0.93	0.93	0.92	0.91	**0.93**

Table 3: Accuracy of Logistic Regression models trained after adding selective data points from the pool as determined by each method. (Bold: highest, Underlined: second highest)

4.3 Importance of DataSift's components

In this set of experiments, we address RQ2 by investigating the respective contributions of the MAB component and the data valuation component within the DataSift framework. Specifically, we compare DataSift against two baselines: (i) DataSift_M, a variant of DataSift, which relies solely on the MAB framework followed by random batch selection instead of data valuation, and (ii) Inf, which utilizes only data valuation (no MAB) for data selection. This comparison underscores the critical importance of integrating both MAB and data valuation in achieving the effectiveness of DataSift. We report performance results—measured in terms of both accuracy and fairness—on three representative datasets (out of six), chosen for their distinct empirical behaviors observed during experiment.

We report logistic regression results across all datasets. Figure 6 follows the standard layout, with the x-axis denoting the budget fraction and the y-axis showing the evaluation metrics. The first column reports fairness outcomes, where values within the green-shaded region (i.e., close to zero) indicate a fair model, while the second column presents the corresponding classification accuracy.

For the **AdultIncome** dataset (Figure 6a), DataSift achieves a zero-bias model by incorporating only 6% of the data pool, whereas DataSift_M requires 20% and, despite substantial improvement, still fails to reach the fairness region. A key limitation of DataSift_M is its reliance on random batch selection within the partition with the highest potential reward. While the partition itself may be promising, not all points within it are beneficial—some can even worsen fairness. Consequently, repeated random sampling causes DataSift_M to perform only marginally better than the Random baseline and other expansion methods (Figure 3b). In contrast, the valuation-based selection in DataSift effectively addresses this issue and drives the model to zero bias. An interesting pattern

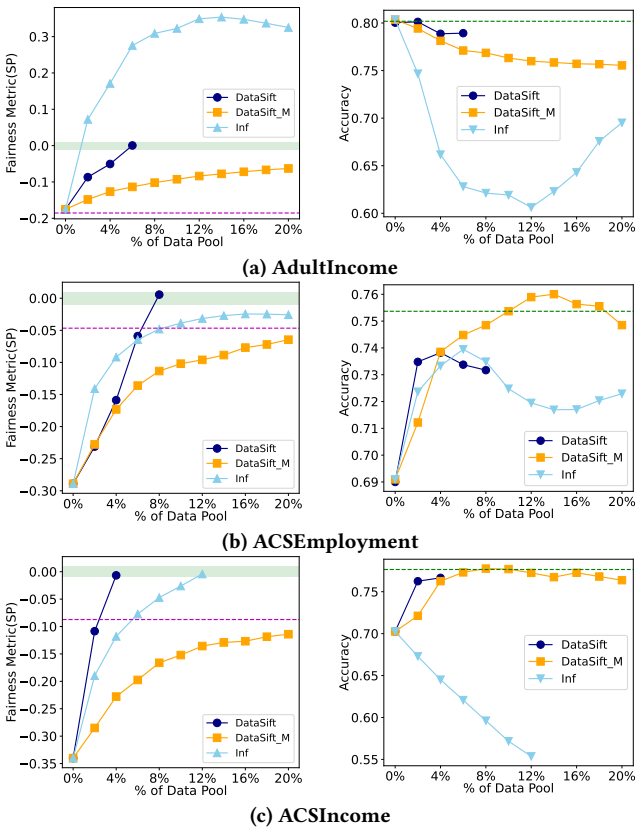


Figure 6: Comparing DataSift with DataSift_M and Inf to highlight the rationale behind DataSift's formation. For each dataset, the left panel shows fairness improvement and the right panel shows accuracy. The magenta and green dotted lines indicate fairness and accuracy, respectively, after adding the entire data pool.

emerges with the Inf method. Although the model initially exhibits bias favoring the privileged group (negative metric values), as Inf continues to exploit the most influential batches from data pool, the model becomes increasingly biased in the opposite direction (positive values). This reversal occurs because the most influential data points often come from the protected group; repeated exploitation of the same information amplifies bias in opposite direction rather than mitigating it. By dynamically balancing exploration and exploitation, DataSift avoids this pitfall and converges to a fair model. Finally, we note that if Inf is stopped early (e.g., at 0–2% expansion), fairness can be temporarily improved, but at the cost of a 13% drop in accuracy (see corresponding accuracy at Figure 6a). By contrast, DataSift consistently preserves accuracy while driving the fairness metric toward zero across the entire expansion process.

For the **ACSEmployment** dataset in Figure 6b, DataSift achieves the fairness goal by incorporating only 8% of the data pool, whereas Inf requires 20% and still fails to reach the fairness region—though it performs better than DataSift_M and the other baselines (Figure 3c). As in earlier cases, DataSift_M cannot achieve full fairness due to its influence-agnostic batch selection, but it marginally outperforms the simpler baselines. For this substantially larger dataset, Inf improves fairness slowly, even when adding the most influential points, because batch interdependencies diminish actual gains (refer to Section 3.3.2). This shows that batch influence is not simply additive, and repeatedly exploiting top-K influential batches yields only marginal improvements. While this strategy enhances fairness to some extent, it does so at the cost of lower accuracy relative to other methods (see accuracy plot). In contrast, when the performance of DataSift declines compared to Inf in the initial 4% of data pool, it transitions to a new partition that provides additional information, resulting in a sharp improvement in fairness in next iteration. Moreover, DataSift gradually improves accuracy beyond its initial level(around 6%), while DataSift_M attains slightly higher accuracy(around 9%) by incorporating more random data points, albeit at the expense of fairness.

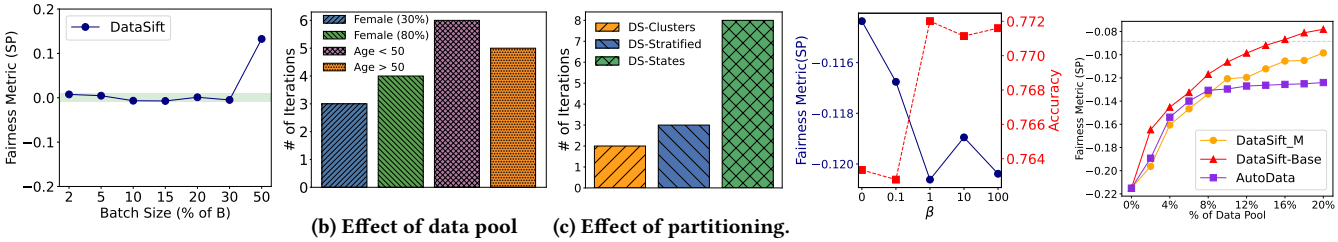
For the **ACSIncome** dataset (Figure 6c), both DataSift and Inf achieve the fairness goal, requiring only 4% and 12% of the data pool, respectively—again highlighting the efficiency of DataSift. While Inf substantially outperforms other baselines in fairness (Figure 3b), it suffers a drastic accuracy loss of nearly 23%. As observed for the **ACSEmployment** dataset, this decline arises from repeatedly exploiting top influential points, which are disproportionately drawn from the protected group. The resulting redundancy improves fairness but fails to enhance predictive power, thereby degrading accuracy. In contrast, DataSift leverages the integration of MAB with data valuation to selectively acquire high-quality data points from unexplored regions of the pool. Consequently, DataSift attains a 10% increase in accuracy while maintaining a fair model. Although DataSift_M exhibits a comparable accuracy gain, its fairness level remains substantially lower.

4.4 Ablation analysis

This section reports the sensitivity of our solutions to the hyperparameters and other design choices. All experiments are conducted on the **ACSIncome** dataset in this section.

Effect of batch size. In our variations of DataSift, we define the batch size as a percentage of the expansion threshold, denoted by B. Figure 7a illustrates the final fairness metric (y-axis) for a logistic regression model trained on selectively expanded data from the **ACSIncome** dataset, while varying the batch size from 2% to 50% of the total budget (x-axis). The results show that the performance of DataSift remains stable across batch sizes of up to 30%, consistently achieving fairness regardless of the specific choice. However, excessively large batch sizes—around 50% of B—lead to a decline in performance, introducing bias in the opposite direction. This degradation occurs because Inf, under large batch settings, selects predominantly influential data points concentrated in the protected group. With only two large acquisitions possible, the framework cannot sufficiently balance these effects, resulting in residual bias. By contrast, smaller batch sizes (up to 30% of B) enable multiple evaluation rounds, allowing DataSift to adjust its selections iteratively and ultimately converge to a fair model.

Effect of data pool distribution. DataSift does not assume that the data pool follows the same distribution as the training data. To substantiate this claim, we construct data pools whose distributions differ markedly from the training set. While the original training data has 48% representing the protected group, and ages spanning



(a) Effect of batch size (b) Effect of data pool distribution (c) Effect of partitioning. (DS:DataSift) (d) Effect of β (e) Reward Score selection

Figure 7: Effect of Hyper-parameters and design choices (ACSIncome dataset).

from 17 to 94, we consider the following four scenarios for the data pool: (a) fraction of protected group = 0.3 (KL Divegence=0.19), (b) fraction of protected group = 0.8 (KL Divegence=0.26), (c) age-restricted data pool 1 (age < 50, KL Divegence=0.39), and (d) age-restricted data pool 2 (age ≥ 50, KL Divegence=0.96). Across all four settings, DataSift consistently attains the target fairness threshold; accordingly, Figure 7b reports only the number of acquisition iterations required to reach the goal. In gender-skewed pools, DataSift requires three and four iterations (compared to two when the pool follows the training distribution), whereas age-restricted pools require five and six iterations. The increased number of iterations in these cases are expected, as the pool size is substantially reduced (one-third of the original), making the finding of influential samples more gradual. Nevertheless, DataSift consistently improves fairness across all settings, while baseline methods degrade under distributional shifts (see Appendix A.2.1 for a detailed comparison). These results confirm that DataSift acquisition is driven by observed utility signals through influence-based candidate subsets of pool, rather than by assumptions of global distributional alignment.

Effect of partitioning. DataSift requires partitioning the data space into smaller groups, which can be accomplished either automatically (e.g., via clustering) or by leveraging domain-specific expertise. In Figure 7c, we evaluate DataSift under three partitioning strategies: (i) clustering into the optimal number of groups using a Gaussian Mixture Model (GMM) (see Appendix A.2.2 for the effectiveness of alternative clustering approaches), (ii) geographic partitioning by state (treating the four largest states as separate partitions), and (iii) stratified partitioning into four subsets defined by the cross-product of a binary sensitive attribute and a binary outcome. Across all three strategies, DataSift successfully achieves a completely fair model, highlights the same effectiveness, so we report the results for the required number of iterations in Figure 7c. DataSift-Cluster and DataSift-Stratified reach fairness with only 2 and 3 iterations, respectively, while DataSift-States requires a substantially larger, 8 iterations. This inefficiency arises because the four largest states in the dataset share highly similar distributions, limiting the diversity of information gained by switching between partitions. By contrast, clustering or stratification creates more heterogeneous partitions, enabling faster convergence to fairness.

Effect of β. In Figure 7d, we analyze the effect of varying β, the parameter that balances fairness and accuracy in the reward function defined in Equation 1. Setting $\beta = 0$ places exclusive emphasis on fairness when shaping the reward distribution, whereas larger values of β progressively shift the focus toward accuracy. As expected,figure 7d shows that increasing β reduces fairness while improving accuracy. Hence, the choice of β should be guided by the requirements of the specific application, with domain expertise playing a central role in determining the appropriate trade-off.

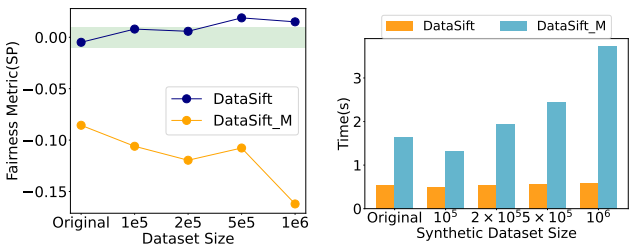
Effect of choice of reward score. DataSift builds on the MAB paradigm, which requires designing a reward score to approximate the reward distribution across partitions. AutoData employs a distance-centric reward optimized for accuracy, whereas DataSift incorporates base-rate differences among sensitive groups into the reward design (Equation 1) to account for both fairness and accuracy. For completeness, we report results for DataSift_M. As shown in Figure 7e, AutoData fails to yield stable fairness improvements. Reward scores based solely on base rates in our framework, DataSift-Base, substantially improve fairness but often sacrifice accuracy. By contrast, the reward function in DataSift_M effectively balances the trade-off, promoting equity and performance.

Implementation and parameter guidelines. While DataSift introduces multiple components, its implementation remains practical, as most parameters have intuitive interpretations and stable defaults. The batch size controls the granularity of data acquisition, the fairness threshold specifies the desired tolerance level, and the exploration parameters follow standard multi-armed bandit practices. In our experiments, a single set of parameters was used across datasets, and in ablation analysis, we observed that performance is robust to moderate parameter variations, indicating limited sensitivity to parameter variations. In practice, we recommend starting from these default settings and adjusting the parameters based on computational budget and fairness requirements.

4.5 Scalability Analysis

This section answers RQ4 that evaluates the different methods with respect to variations in dataset and data pool sizes.

Using the **AdultIncome** dataset, we synthetically generate additional data points to vary the pool size from the original dataset up to 10^6 records. Throughout this study, the budget and batch size remain fixed. Figure 8a reports the fairness achieved by logistic regression models trained with DataSift and DataSift_M under



(a) Effectiveness (b) Efficiency

Figure 8: Scalability on synthetic dataset.

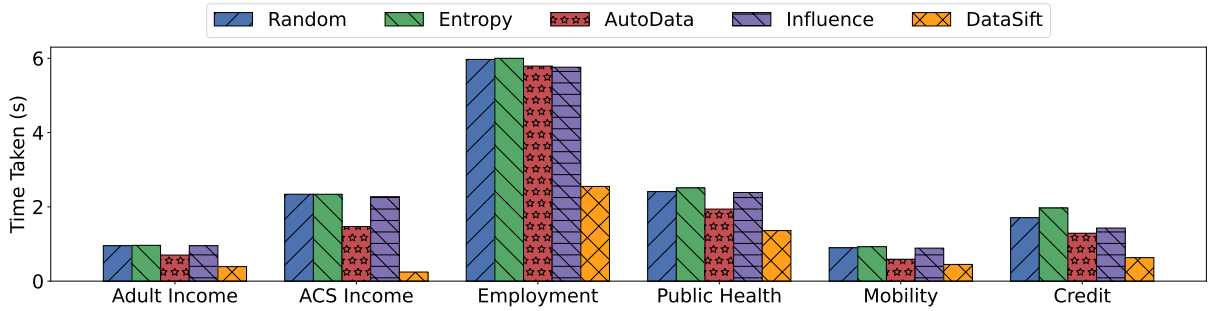


Figure 9: Efficiency on real-world datasets.

different pool sizes. The y-axis represents the ultimate fairness achieved by the model under various scenarios. The results show that DataSift maintains nearly consistent performance as the data pool grows, whereas DataSift_M experiences a slight decline. Unlike DataSift_M, which relies on random batch selection and is thus influenced by the data pool's distribution, DataSift consistently identifies influential points irrespective of pool size, leading to more stable fairness outcomes.

Efficiency analysis. Figure 9 reports the runtime of the data expansion framework, excluding all pre-computation costs for all methods. All the baselines typically consume the entire expansion budget, evaluating a large number of candidate batches. Since each additional batch evaluation requires retraining the model, this results in substantially higher computational cost. In contrast, DataSift incurs a modest overhead for reward updates and influence-based ranking, but achieves significantly lower overall runtime by terminating early once the fairness objective is satisfied.

Figure 8b evaluates scalability under increasing data pool sizes using a synthetically enlarged AdultIncome dataset. For DataSift, batch construction cost remains nearly constant since influence scores are computed once on the fixed training set and the top-K influential points are selected deterministically. As a result, runtime is largely insensitive to pool size. In contrast, DataSift_M relies on random batch construction, making its runtime grow with the size of the data pool.

5 RELATED WORK

This work intersects four research areas: algorithmic fairness, data expansion/acquisition, reinforcement learning, and data valuation. While each of these areas has been studied extensively, our work is novel in integrating data valuation with a multi-armed bandit framework to selectively acquire high-quality data for improving model fairness.

Algorithmic fairness. As machine learning systems are increasingly deployed in high-stakes domains such as criminal justice, healthcare, and finance, fairness violations have become a critical concern [10, 19, 30, 34]. Existing bias mitigation techniques are typically categorized as pre-processing, in-processing, or post-processing methods [14, 41]. Among these, pre processing approaches are particularly attractive due to their simplicity and model-agnostic nature [32]. Our work falls into this category by improving fairness through selective modification of the training data.

Data expansion/acquisition for model fairness. Data quality has long been a central concern in data management [52, 60, 64], and its role in improving machine learning performance has recently gained attention [3, 15, 37, 38, 63]. We focus on *data expansion* [21], which augments training data by selectively acquiring additional high-quality examples. Unlike prior work that primarily targets accuracy or data augmentation for vision tasks [56], our method explicitly targets model fairness. Our approach is most closely related to AutoData [15], which uses a bandit-based strategy to acquire data for improving accuracy; however, AutoData relies on distance-based clustering and random batch selection, making it unsuitable for fairness objectives. In contrast, DataSift leverages data valuation to guide fairness-aware acquisition.

Reinforcement learning in data management. Reinforcement learning techniques have been increasingly applied to data management problems, including data cleaning, integration, and acquisition [6, 12, 15, 31, 58]. Multi-armed bandits, in particular, offer efficient exploration–exploitation trade-offs and have been shown to achieve near-optimal solutions with limited feedback. While AutoData [15] also adopts a bandit framework, its reward formulation is accuracy-centric and does not address fairness.

Data valuation. Data valuation techniques, including influence functions [35] and data Shapley [29], provide principled ways to measure the contribution of individual data points and have recently been applied in data management [7, 22, 40]. While data Shapley is model-agnostic but computationally expensive, influence functions offer efficient first-order approximations with a one-time offline cost, which we leverage for fairness-aware data acquisition.

6 CONCLUSIONS AND FUTURE WORK

We present a novel approach for solving the problem of data expansion to improve machine learning model fairness by determining which data points in a data pool must be added to the underlying training data. We introduce DataSift, a principled framework that integrates reinforcement learning with data valuation to determine the most valuable data points to incorporate. We demonstrate experimentally that data valuation is crucial in effective discovery of useful data points and that DataSift is both effective and efficient in identifying data points valuable for model fairness. In the future, we plan to explore several interesting directions: (a) expand the data valuation module to efficiently cater to non-parametric ML algorithms. (b) explore alternate reinforcement learning approaches (e.g., Q-learning) and other exploration-exploitation trade-off algorithms (e.g., Thompson sampling) since the MAB-based framework is limited by the choice of identified partitions. Other directions include incorporating the cost of data acquisition in the reward scores and extending the framework to the task of source discovery.

ARTIFACTS

The experiments run on a MacBook Pro (Apple M3 Pro, 36GB LPDDR5) using Python 3.11.5 in Jupyter Notebook. The ready-to-run source code for DataSift is available at this link: DataSift.

REFERENCES

[1] Shipra Agrawal and Navin Goyal. 2012. Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*. JMLR Workshop and Conference Proceedings, Edinburgh, Scotland, 39–1.

[2] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. *Machine Bias: There's Software Used Across the Country to Predict Future Criminals. And It's Biased Against Blacks.* ProPublica, May 23, 2016.

[3] Abolfazl Asudeh, Zhongjun Jin, and H. V. Jagadish. 2019. Assessing and Remedying Coverage for a Given Dataset. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, Macau, China, 554–565. https://doi.org/10.1109/ICDE.2019.00056 ISSN: 2375-026X.

[4] Peter Auer. 2000. Using upper confidence bounds for online learning. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, IEEE, Redondo Beach, CA, USA, 270–279.

[5] Peter Auer. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3, Nov (2002), 397–422.

[6] Laure Berti-Equille. 2019. Learn2clean: Optimizing the sequence of tasks for web data preparation. In *The world wide web conference*. 2580–2586.

[7] Leopoldo Bertossi, Benny Kimelfeld, Ester Livshits, and Mikaël Monet. 2023. The Shapley value in database management. *ACM Sigmod Record* 52, 2 (2023), 6–17.

[8] Sumon Biswas and Hridesh Rajan. 2021. Fair preprocessing: towards understanding compositional fairness of data transformers in machine learning pipeline. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 981–993.

[9] Sumon Biswas, Mohammad Wardat, and Hridesh Rajan. 2022. The art and practice of data science pipelines: A comprehensive study of data science pipelines in theory, in-the-small, and in-the-large. In *Proceedings of the 44th International Conference on Software Engineering*. 2091–2103.

[10] Brakkton Booker. 2019. Housing Department Slaps Facebook With Discrimination Charge. lhttps://www.npr.org/2019/03/28/707614254/hud-slaps-facebook-with-housing-discrimination-charge.

[11] Sébastien Bubeck, Nicolo Cesa-Bianchi, and Gábor Lugosi. 2013. Bandits with heavy tail. *IEEE Transactions on Information Theory* 59, 11 (2013), 7711–7717.

[12] Qingpeng Cai, Can Cui, Yiyuan Xiong, Wei Wang, Zhongle Xie, and Meihui Zhang. 2023. A Survey on Deep Reinforcement Learning for Data Processing and Analytics. *IEEE Transactions on Knowledge and Data Engineering* 35, 5 (2023), 4446–4465. https://doi.org/10.1109/TKDE.2022.3155196

[13] Alexandra Carpentier, Alessandro Lazaric, Mohammad Ghavamzadeh, Rémi Munos, and Peter Auer. 2011. Upper-confidence-bound algorithms for active learning in multi-armed bandits. In *International Conference on Algorithmic Learning Theory*. Springer, 189–203.

[14] Simon Caton and Christian Haas. 2024. Fairness in Machine Learning: A Survey. *ACM Comput. Surv.* 56, 7, Article 166 (April 2024), 38 pages. https://doi.org/10.1145/3616865

[15] Chengliang Chai, Jiabin Liu, Nan Tang, Guoliang Li, and Yuyu Luo. 2022. Selective data acquisition in the wild for model charging. *Proc. VLDB Endow.* 15, 7 (March 2022), 1466–1478. https://doi.org/10.14778/3523210.3523223

[16] Alexandra Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *CoRR* abs/1703.00056 (2017).

[17] R. Dennis Cook and Sanford Weisberg. 1980. Characterizations of an Empirical Influence Function for Detecting Influential Cases in Regression. *Technometrics* 22, 4 (1980), 495–508.

[18] Will Cukierski Credit Fusion. 2011. Give Me Some Credit. https://kaggle.com/competitions/GiveMeSomeCredit

[19] Jeffrey Dastin. 2018. RPT-INSIGHT-Amazon scraps secret AI recruiting tool that showed bias against women. *Reuters* (2018).

[20] G. Davis, S. Mallat, and M. Avellaneda. 1997. Adaptive Greedy Approximations. *Constructive Approximation* 13, 1 (1997), 57. https://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=a9h&AN=8864613&site=ehost-live&custid=purdue

[21] Yihe Deng, Yu Yang, Baharan Mirzasoleiman, and Quanquan Gu. 2023. Robust Learning with Progressive Data Expansion Against Spurious Correlation. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 1390–1402. https://proceedings.neurips.cc/paper_files/paper/2023/file/0506ad3d1bcc8398a920db9340f27fe4-Paper-Conference.pdf

[22] Daniel Deutch, Nave Frost, Amir Gilad, and Oren Sheffer. 2021. Explanations for Data Repair Through Shapley Values. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (Virtual Event, Queensland, Australia) *(CIKM '21)*. Association for Computing Machinery, New York, NY, USA, 362–371. https://doi.org/10.1145/3459637.3482341

[23] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring adult: New datasets for fair machine learning. *Advances in neural information processing systems* 34 (2021), 6478–6490.

[24] Dheeru Dua, Casey Graff, et al. 2017. UCI machine learning repository. (2017).

[25] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, Vol. 96. 226–231.

[26] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A data discovery system. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 1001–1012.

[27] Mario A. T. Figueiredo and Anil K. Jain. 2002. Unsupervised learning of finite mixture models. *IEEE Transactions on pattern analysis and machine intelligence* 24, 3 (2002), 381–396.

[28] Aurélien Garivier and Eric Moulines. 2011. On upper-confidence bound policies for switching bandit problems. In *International conference on algorithmic learning theory*. Springer, 174–188.

[29] Amirata Ghorbani and James Zou. 2019. Data Shapley: Equitable Valuation of Data for Machine Learning. In *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97. PMLR, 2242–2251.

[30] Karen Hao. 2019. Self-driving cars more likely to hit blacks. lhttps://www.technologyreview.com/2019/03/01/136808/self-driving-cars-are-coming-but-accidents-may-not-be-evenly-distributed/.

[31] Yuval Heffetz, Roman Vainshtein, Gilad Katz, and Lior Rokach. 2020. DeepLine: AutoML Tool for Pipelines Generation using Deep Reinforcement Learning and Hierarchical Actions Filtering. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 2103–2113. https://doi.org/10.1145/3394486.3403261

[32] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé, Miro Dudik, and Hanna Wallach. 2019. Improving Fairness in Machine Learning Systems: What Do Industry Practitioners Need?. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–16.

[33] Anil K Jain. 2010. Data clustering: 50 years beyond K-means. *Pattern recognition letters* 31, 8 (2010), 651–666.

[34] Matthew Kay, Cynthia Matuszek, and Sean A. Munson. 2015. Unequal Representation and Gender Stereotypes in Image Search Results for Occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3819–3828.

[35] Pang Wei Koh and Percy Liang. 2017. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70. 1885–1894.

[36] Volodymyr Kuleshov and Doina Precup. 2014. Algorithms for multi-armed bandit problems. *arXiv preprint arXiv:1402.6028* (2014).

[37] Yifan Li, Xiaohui Yu, and Nick Koudas. 2021. Data Acquisition for Improving Machine Learning Models. *Proceedings of the VLDB Endowment* 14, 10 (June 2021), 1832–1844. https://doi.org/10.14778/3467861.3467872 arXiv:2105.14107 [cs].

[38] Yifan Li, Xiaohui Yu, and Nick Koudas. 2024. Data Acquisition for Improving Model Confidence. *Proc. ACM Manag. Data* 2, 3, Article 131 (May 2024), 25 pages. https://doi.org/10.1145/3654934

[39] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just Train Twice: Improving Group Robustness without Training Group Information. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 6781–6792. https://proceedings.mlr.press/v139/liu21f.html

[40] Xuan Luo and Jian Pei. 2024. Applications and computation of the Shapley value in databases and machine learning. In *Companion of the 2024 International Conference on Management of Data*. 630–635.

[41] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2022. A Survey on Bias and Fairness in Machine Learning. *Comput. Surveys* 54, 6 (July 2022), 1–35. https://doi.org/10.1145/3457607

[42] Asit Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. 2021. Accelerating sparse deep neural networks. *arXiv preprint arXiv:2104.08378* (2021).

[43] Fionn Murtagh and Pedro Contreras. 2012. Algorithms for hierarchical clustering: an overview. *Wiley interdisciplinary reviews: data mining and knowledge discovery* 2, 1 (2012), 86–97.

[44] Fatemeh Nargesian, Abolfazl Asudeh, and HV Jagadish. 2021. Tailoring data source distributions for fairness-aware data integration. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2519–2532.

[45] Fatemeh Nargesian, Ken Q Pu, Erkang Zhu, Bahar Ghadiri Bashardoost, and Renée J Miller. 2020. Organizing data lakes for navigation. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1939–1950.

[46] Fatemeh Nargesian, Erkang Zhu, Ken Q Pu, and Renée J Miller. 2018. Table union search on open data. *Proceedings of the VLDB Endowment* 11, 7 (2018), 813–825.

[47] Andrew A Neath and Joseph E Cavanaugh. 2012. The Bayesian information criterion: background, derivation, and applications. *Wiley Interdisciplinary Reviews:*

Computational Statistics 4, 2 (2012), 199–203.
[48] Nhan Nguyen-Thanh, Dana Marinca, Kinda Khawam, David Rohde, Flavian Vasile, Elena Simona Lohan, Steven Martin, and Dominique Quadri. 2019. Recommendation system-based upper confidence bound for online advertising. *arXiv preprint arXiv:1909.04190* (2019).
[49] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
[50] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.
[51] Neoklis Polyzotis and Matei Zaharia. 2021. What can Data-Centric AI Learn from Data and ML Engineering? *CoRR* abs/2112.06439 (2021). arXiv:2112.06439 https://arxiv.org/abs/2112.06439
[52] Theodoros Rekatsinas, Amol Deshpande, Xin Luna Dong, Lise Getoor, and Divesh Srivastava. 2016. Sourcesight: Enabling effective source selection. In *Proceedings of the 2016 International Conference on Management of Data*. 2157–2160.
[53] Nima Shahbazi, Mahdi Erfanian, and Abolfazl Asudeh. 2024. Coverage-based Data-centric Approaches for Responsible and Trustworthy AI. *IEEE Data Engineering Bulletin* (2024).
[54] Nima Shahbazi, Yin Lin, Abolfazl Asudeh, and HV Jagadish. 2023. Representation bias in data: A survey on identification and resolution techniques. *Comput. Surveys* 55, 13s (2023), 1–39.
[55] C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal* 27, 3 (1948), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x
[56] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of big data* 6, 1 (2019), 1–48.
[57] Aleksandrs Slivkins et al. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning* 12, 1-2 (2019), 1–286.
[58] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
[59] Ki Hyun Tae and Steven Euijong Whang. 2021. Slice Tuner: A Selective Data Acquisition Framework for Accurate and Fair Machine Learning Models. In *Proceedings of the 2021 International Conference on Management of Data* (Virtual Event, China) *(SIGMOD '21)*. Association for Computing Machinery, New York, NY, USA, 1771–1783. https://doi.org/10.1145/3448016.3452792
[60] Ki Hyun Tae and Steven Euijong Whang. 2021. Slice tuner: A selective data acquisition framework for accurate and fair machine learning models. In *Proceedings of the 2021 International Conference on Management of Data*. 1771–1783.
[61] Sahil Verma and Julia Rubin. 2018. Fairness definitions explained. In *Proceedings of the international workshop on software fairness*. Association for Computing Machinery, 1–7.
[62] Joannes Vermorel and Mehryar Mohri. 2005. Multi-armed bandit algorithms and empirical evaluation. In *European conference on machine learning*. Springer, 437–448.
[63] Tingting Wang, Shixun Huang, Zhifeng Bao, J Shane Culpepper, Volkan Dedeoglu, and Reza Arablouei. 2024. Optimizing Data Acquisition to Enhance Machine Learning Performance. *Proceedings of the VLDB Endowment* 17, 6 (2024), 1310–1323.
[64] Gerhard Weikum. 2013. Data discovery. *Data Science Journal* 12 (2013).
[65] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. 2023. Data collection and quality challenges in deep learning: a data-centric AI perspective. *The VLDB Journal* 32, 4 (Jan. 2023), 791–813. https://doi.org/10.1007/s00778-022-00775-9
[66] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, and Xia Hu. [n. d.]. *Data-centric AI: Perspectives and Challenges*. 945–948. https://doi.org/10.1137/1.9781611977653.ch106 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611977653.ch106
[67] Daochen Zha, Kwei-Herng Lai, Fan Yang, Na Zou, Huiji Gao, and Xia Hu. 2023. Data-centric AI: Techniques and Future Perspectives. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) *(KDD '23)*. Association for Computing Machinery, New York, NY, USA, 5839–5840. https://doi.org/10.1145/3580305.3599553
[68] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1997. BIRCH: A new data clustering algorithm and its applications. *Data mining and knowledge discovery* 1, 2 (1997), 141–182.

A APPENDIX

A.1 Generalizability across fairness metrics

Algorithmic group fairness requires that individuals belonging to different demographic groups be treated comparably by a decision-making system. Several formal metrics have been proposed to quantify such notions of fairness, including statistical (demographic) parity, true positive rate (TPR) parity, predictive parity, and equalized odds. While the main body of this paper focuses on statistical parity—under which DataSift consistently outperforms the baselines—we emphasize that the proposed framework is not limited to a single fairness definition. We report results for the following metrics:

True Positive Rate (TPR) Parity. We first consider *true positive rate parity* (also known as *equal opportunity*), which requires equal true positive rates across groups. Formally, TPR parity is satisfied if

$$\Pr(\hat{Y} = 1 \mid Y = 1, A = a) = \Pr(\hat{Y} = 1 \mid Y = 1, A = b),$$

for all protected groups $A \in \{a, b\}$. This metric focuses on ensuring that qualified individuals have equal chances of receiving a positive outcome regardless of group membership. Figure 10 reports the corresponding results. DataSift outperforms all the baselines across datasets–reach the fairness goal within less number of iterations exploiting only 4 to 8% of data pool.

Predictive Parity. We further evaluate DataSift under *predictive parity*, which requires that a model's precision be equal across groups. Formally, predictive parity holds if, for all protected groups $A \in \{a, b\}$,

$$\Pr(Y = 1 \mid \hat{Y} = 1, A = a) = \Pr(Y = 1 \mid \hat{Y} = 1, A = b),$$

where Y denotes the true label and $\hat{Y}$ denotes the predicted label. This criterion ensures that, among individuals receiving a positive prediction, the likelihood of being correctly classified is the same across groups. Figure 11 reproduces the main experimental setting of Figure 3 using predictive parity as the fairness objective for two datasets. For the remaining datasets, the initial training data already meets the predefined fairness threshold, and thus the expansion process is omitted. The results demonstrate that DataSift remains effective in reducing group-level disparities under this metric. As

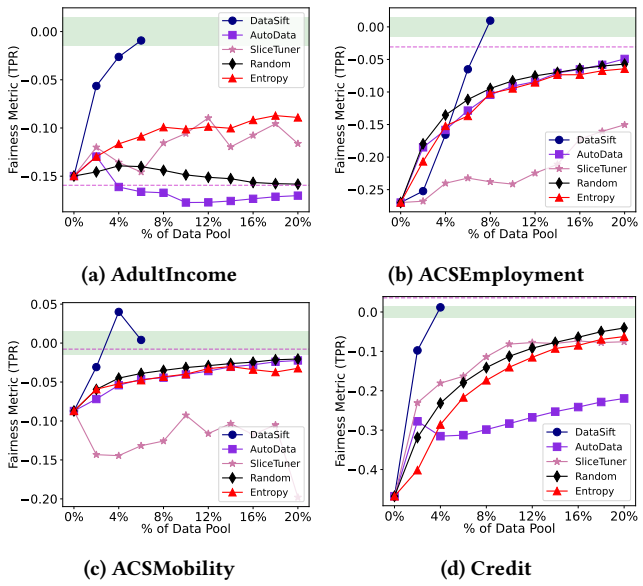
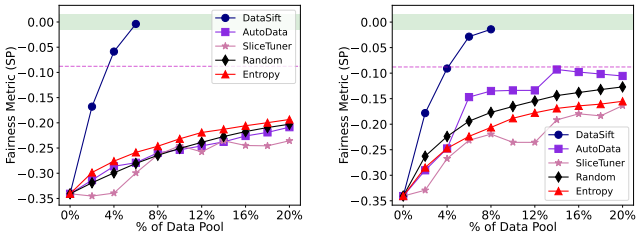
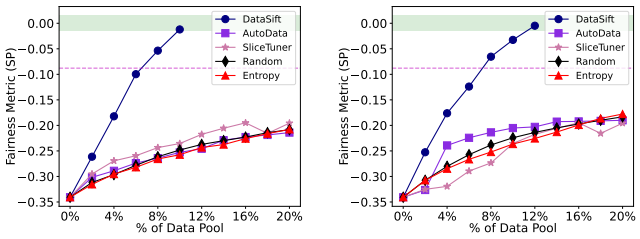


Figure 10: DataSift vs. baselines (True Postive Rate Parity).



(a) Protected fraction = 0.3, KL divergence = 0.19

(b) Protected fraction = 0.8, KL divergence = 0.26



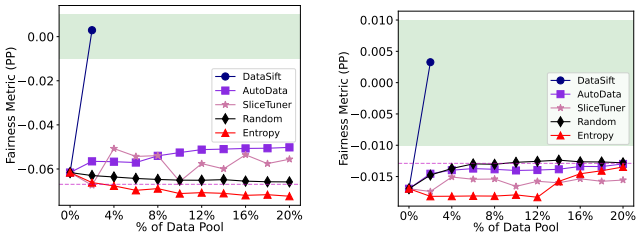
(c) Age > 50, KL divergence = 0.96

(d) Age < 50, KL divergence = 0.39

Figure 12: Varying data pool distributions.

with TPR parity , DataSift consistently improves fairness under predictive parity while maintaining competitive predictive performance.

Taken together, these additional experiments confirm that the effectiveness of DataSift is not tied to a specific fairness metric, rather it works as like a plug-and-play method. By leveraging reward-driven partition selection and influence-based acquisition, the framework generalizes naturally across different definitions of group fairness, including outcome-based and error-rate–based criteria. This highlights the flexibility of DataSift and its applicability in settings where different fairness notions may be required.



(a) ACSEmployment

(b) ACSMobility

Figure 11: DataSift vs. baselines (Predictive Parity).

A.2 Ablation Analysis

A.2.1 Datapool Distribution. To evaluate whether the proposed framework, DataSift the data pool to follow the same distribution as the initial training data, we conduct an additional sensitivity analysis by deliberately introducing distributional mismatches between the training set and the pool. In the original training data, the fraction of female samples is approximately 0.48. We construct two alternative pool distributions by fixing the female fraction to 0.3 and 0.8, respectively, while keeping the training data unchanged. The corresponding results are shown in Appendix Figures 12a and 12b.

Across both scenarios, DataSift consistently reduces the statistical parity gap and attains the target fairness threshold by selectively acquiring at most 10% of the data pool, even when the pool distribution is substantially skewed relative to the training data. In contrast, the baseline methods exhibit slower convergence or unstable behavior under these distribution shifts. Notably, when the representation of the protected group is severely limited (Figure 12a), the baselines struggle to identify fairness-improving samples, whereas DataSift is able to extract informative instances even from sparsely represented subpopulations. These results demonstrate that DataSift does not depend on the data pool matching the training distribution, but instead dynamically adapts its acquisition strategy to the current model state.

We further assess robustness to shifts in non-sensitive attributes by varying the age distribution of the data pool. In particular, we consider two extreme scenarios in which the pool consists exclusively of individuals older than 50 or exclusively of individuals younger than 50, while the training data spans ages from 17 to 94. The corresponding results are reported in Appendix Figures 12c and 12d. Consistent with the gender-based experiments, DataSift continues to achieve steady reductions in the fairness gap under both settings, demonstrating robustness to substantial changes in the pool's feature composition.

Overall, these experiments confirm that DataSift does not assume the data pool and training data share the same underlying distribution. Instead, the method leverages reward-driven partition selection and influence-based acquisition to remain effective even when the pool is biased, skewed, or otherwise distributionally different from the initial training set.

A.2.2 Clustering sensitivity. Partitioning the data pool serves two key purposes in DataSift framework: it enables systematic exploration of fairness-relevant data compositions and reduces computational overhead by restricting selection to smaller, structured subsets. We report an ablation over alternative pool clustering methods (GMM, K-means, hierarchical-BIRCH, and DBSCAN) with the ACSIncome dataset in Appendix Figure 13.

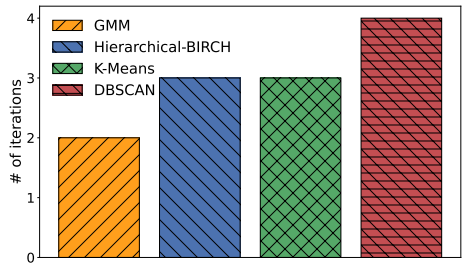


Figure 13: Effect of clustering methods on DataSift

Across all clustering methods, DataSift consistently achieves the target fairness threshold of ±0.1, whereas the considered baselines fail to reach this threshold, consistent with the trends observed in Figure 3b. Accordingly, we report only the number of

acquisition iterations (i.e., the number of selected mini-batches) required by DataSift to attain the fairness goal under each clustering method in Figure 13. Overall, the performances are consistent across GMM, K-means, and BIRCH: the proposed bandit-driven expansion achieves comparable fairness improvements with only minor differences in convergence speed. This indicates that our approach is largely partition-agnostic as long as the pool is split into a small number of stable, coarse-grained groups. DBSCAN requires a slightly higher number of iterations, which is expected since density-based clustering tends to yield many small clusters, increasing the exploration space and introducing higher variance in the reward estimates. These results indicate that simple and scalable partitioning methods (e.g., GMM, K-means, and BIRCH) are sufficient in practice; nevertheless, DataSift is not restricted to any specific clustering strategy.